

版本测试过程优秀实践

Ai学：李志



自我介绍



姓名：李志

Testerhome帐号：arthur

测试经验：10年以上

华为无线产品线，6年多

携程金融，3年

学霸君Ai学，1年

1

版本控制方面优秀实践

2

测试过程优秀实践

3

团队效率提升方面优秀实践

4

线上问题处理方面优秀实践

优秀实践1：迭代发布控制

确认上线时间

开发转测试时间需要在PRD评审完成后，第二天与测试共同确认转测试时间以及正式上线时间。

每天迭代一次

所有版本在转测试开始之后，采用每天一迭代的转测试版本模式，不允许额外的bug修改造成的临时发布和打包。

超纲发布特批

如果遇到阻塞性问题确实需要重新发布和打包，需要上报过研发总监，说明原因之后，经过批准再打包发版。

每天10点发版

从每个迭代版本开始，每天早上10点前，开发邮件确认已经准备好当天测试的版本以及确认合入修改的bug。

迭代发布控制确保版本合入的风险降低

优秀实践2：bug修改控制

依据状态回归

测试以JIRA上bug状态改为Resolved为开始回归的前提，未改状态的bug，测试不进行回归。

Bug提前处理

所有bug在发布前2天的晚上都需要解决完成，剩下1天时间给测试进行QA环境bug回归和QA环境回测，以及发布测试报告。

Bug打回升级

开发第一次修改bug未回归测试通过，在打回的同时，对bug进行升级处理，升级为严重等级，提高大家对修改bug的重视程度。

带bug可发布

在预发布环境当天发现的问题或者发布当天发现的问题，可以经过与研发总结确认，拍板是否需要修改发布，建议非严重问题不再发布。

Bug修改控制确保测试交付顺利

优秀实践3：需求合入控制

转测试不合需求

在版本转测试之后，不再接受新需求的合入。如果有特殊情况，需要研发总监协调，同时测试接受在可推迟发布时间的情况下，接受合入需求。

产品可提bug

如果确认为产品未考虑清楚的情况或者产品设计阶段未充分沟通运营，导致场景遗漏，可以给产品经理提交bug，走bug处理流程。

需求合入控制确保测试的专注性

- 1 版本控制方面优秀实践
- 2 测试过程优秀实践
- 3 团队效率提升方面优秀实践
- 4 线上问题处理方面优秀实践

优秀实践1：推行测试设计

前期方案设计

在完成PRD评审之后，测试进行相关的测试方案设计，利用边界值、等价类加上正交分析的方法，输出完整的测试覆盖内容，用来指导后续的测试设计，同时剔除一些无效的场景和数据覆盖。

完善用例设计

根据方案设计，进行测试用例的设计，用例设计需要步骤清晰，对应的预期结果明确，如果需要前期的预置条件，预置条件也描述清楚。做到用例设计可以给任何人都可以执行。

例行测试评审

测试评审要每次进行，用例确认测试的理解与产品和开发同步，同时场景覆盖要和开发确认已经覆盖完全。

知识地图梳理

针对当前业务场景，梳理出一套完整的业务内容，针对每一个小交互和场景，都定义不同的简称。让团队成员对照知识地图，认清自己对当前业务的覆盖有多少。同时，根据知识地图，补齐测试经验总结文档，让团队所有人都可以快速熟悉业务、避免一些已知的问题。

推行测试设计来确保测试覆盖的全面性

优秀实践2：严格评审制度

开发评审，参与者：开发\测试\版本经理

- 1) SRS评审
- 2) HLD评审
- 3) LLD评审
- 4) bug修改方案评审

【完成后】所有文档归档保存

决策类评审，参与者：SE\TSE\版本经理\测试经理\开发经理

- 1) 每个版本迭代需求排期评审
- 2) 版本迭代bug修改时间评审
- 3) 上线前，决策评审当前需要修改的bug和可以不修改的
- 4) 重大线上问题解决方案评审

测试评审，参与者：开发\测试\版本经理

- 1) 测试需求分析方案评审
- 2) 测试方案评审
- 3) 测试用例评审
- 4) bug测试用例评审

【完成后】所有文档归档保存

评审保证开发和测试的方向和质量的正确性

开发必须组织Code-Review

- 何时组织：在代码Check-in之前
- 参与者：开发经理、周边相关开发、测试
- 怎么做：
 - 开发讲解自己的开发思路
 - 浏览代码结构和调用关系
 - 确认代码规范性
 - 确认代码引用无问题
 - 确认经常踩得坑可以避免

Code-Review保证开发思路和规范，避免继续入坑

优秀实践4：严谨代码对比

参与者：开发

适合版本：在研版本

- **实现方式**：采用自动化工具；
- **对比目的**：查看本次提交内容是否覆盖版本规划的全部需求和修改项目；
- **对比时间**：转测试之后，开始测试之前

参与者：测试

适合版本：上线版本

- **实现方式**：采用自动化工具；
- **对比目的**：查看本次修改是否有其他未涉及的无关代码和异常代码的提交；
- **对比时间**：转测试之后，开始测试之前

代码对比防止修改被遗漏、需求未提交

- 1 版本控制方面优秀实践
- 2 测试过程优秀实践
- 3 团队效率提升方面优秀实践
- 4 线上问题处理方面优秀实践

优秀实践1：重视新员工培养

- 1) **导师责任制**---指定技术专家或者业务专家做导师
- 2) **培养有计划**---导师负责指定3个月的学习计划，包括业务学习，技术学习，工作具体内容和交付件
- 3) **团队要培养**---团队内部各专家每周定期做内部业务和技术培训
- 4) **月度有答辩**---邀请开发和测试相关人员，对员工掌握知识和技能做检视，给员工正确的学习建议和高效的学习方法
- 5) **转正要考核**---邀请各开发和测试领导和专家，对于新员工，利用转正考核的机会，给予后面业务交付和技术提升的指引，帮助他走正确的路，协助他站在巨人的肩膀上

帮助新员工站在巨人的肩膀上

优秀实践2：鼓励开展QCC

QCC是什么：

QCC是英文Quality Control Circles的缩写，中文名质量控制圈。其特点是由基层员工组成的小组，通过适当的训练及引导，使小组能通过定期的会议，去发掘、分析及解决日常工作有关的问题。

QCC怎么用：

由感兴趣的相关人员自动自发的组成QCC圈主动去解决这个问题，基本包括主题选定、制定活动计划、现状调查、目标设定、原因分析、要因确认、对策制定、实施、效果评价、标准化这10步。

QCC有什么用：

QCC圈可以帮助开发、测试和交付能力方面及时发现短板和低效的地方，鼓励员工自发的去自我想办法提升。不限于开发流程、测试流程等，可以考虑交付的各个方面。

QCC组织控制：

- 1) 建立圈要部门专家评审才能通过
- 2) 组织鼓励自我效率提升，给予物质奖励，金额从500到5000不等
- 3) 中期组织会进行检视，确定做的方向和当初一致，效果一致
- 4) 项目要看到成果，才会给予奖励，所以周期从三个月到半年不等，并且要有事实，用数据说话

QCC可以更好的做组织自发的效率提升

优秀实践3：全员参与KM建设

KM基本认识

知识管理(KNOWLEDGE MANAGEMENT)。企业在长期管理运营中,有大量的关于企业的技术、标准、规范、管理、生产、经验等关系企业发展的知识.这些知识有的在技术档案中,有的知识分散在企业各类文件制度条例中,更有大量的知识在企业领导、工人、以及技术人员的头脑中和实践经验中。这是企业的财富——知识财富。

KM带来的好处

- 新员工培养效率提升
- 部门经验持续沉淀
- 技术专家更容易培养
- 人人爱分享，大家爱总结的氛围形成
- 员工业务知识提升，跨平台和领域的问题较少明显

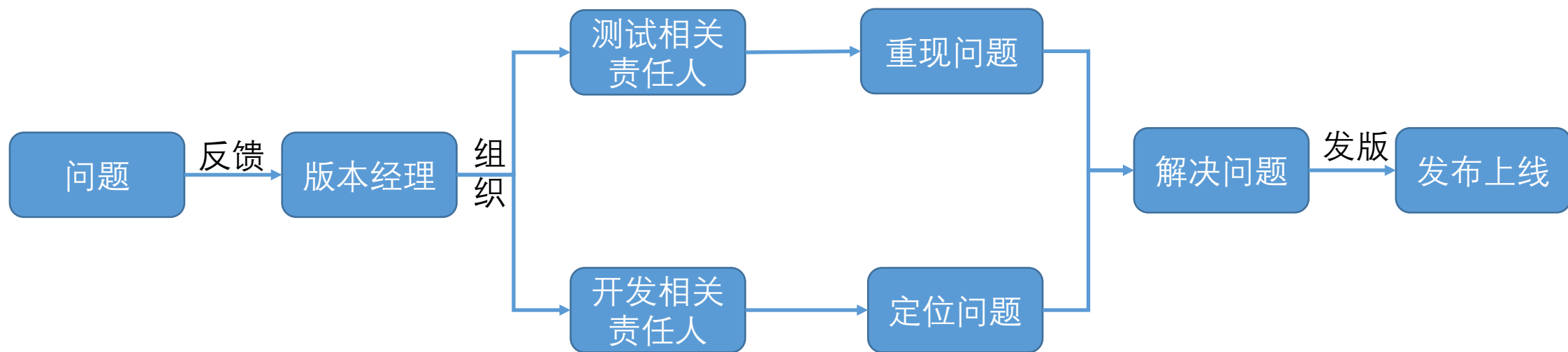
KM怎么落地：

- 1) 采用社区+论坛的方式，提升用户活跃度和黏性，增加各种交互体验
- 2) 平时的经验分享和知识分享，都移植到KM社区
- 3) 历史的经验文档和总结文档复制到社区
- 4) 用例评审，产品评审，回顾会议，开发review等过程文档保存到社区
- 5) 技术性文档保存到社区
- 6) 各种分享类活动通过社区报名
- 7) 组织的活动照片和视频保存到社区
- 8) 社区组织各种技术或者业务或者学习的比赛
- 9) 部门的体育活动组织和照片分享移植到社区
- 10) 找一个积极为社区服务的管理贡献者

KM建设增加知识的沉淀和积累

- 1 版本控制方面优秀实践
- 2 测试过程优秀实践
- 3 团队效率提升方面优秀实践
- 4 线上问题处理方面优秀实践

优秀实践1：线上问题责任制



明确责任和相应机制：

- 1) 测试责任人和开发责任人都是相关问题领域或者模块最熟悉的人。
- 2) 要树立认识线上问题 责任制的规定，要大家第一时间相应。
- 3) 先解决问题，再讨论其他业务交付计划和排期。

线上问题保证第一时间相应和解决

线上问题回溯关键点

- 发起人：
- 版本经理；
- 参与人：
- 版本经理、开发经理、测试经理、开发专家SE、测试专家TSE、开发负责人、测试负责人
- 回溯过程：
 - 1) 从问题发现、定位、测试、解决、上线的过程整体路演，确认问题第一时间被定位和解决，找到其中是否有改进和提升的地方；
 - 2) 回溯开发过程线上问题遗漏的点，提出开发改进项，要求可落地的内容；
 - 3) 回溯测试过程线上问题漏测的点，提出测试改进项，要求可落地的内容；
 - 4) 如果感觉无改进或者改进项目无法落地，则回溯结果为无改进。
- 回溯报告归档：
- 回溯报告统一归档，3个月为一个周期，学习之前的经验和总结教训。

回溯让每一个人对线上问题都特别重视和谨慎

有什么想说的，可以抛出来？

智能教学， 爱上学习

Thank You

